SYSTEMS, METHODS AND SOFTWARE FOR REMOTE PASSWORD AUTHENTICATION USING MULTIPLE SERVERS

ABSTRACT

Systems, methods and software employ zero-knowledge password (ZKP) protocols to provide strong authentication using low-grade passwords that people can easily memorize. We describe protocols that enable multiple servers to verify a password, without providing any single server, client, or network attacker with the ability to validate guesses for the password off-line. Further improvements include removing dependency on a prior secure channel and client-stored keys or certificates, increasing performance without introducing new cryptographic assumptions, and better management of mistakes in password entry. To enroll, a user chooses a password and constructs a master key K composed of multiple shares. The master key may be used for a variety of purposes, such as encrypting the user's private keys and other sensitive data. A set of random values $\{y_1, y_2, \ldots y_N\}$ is selected, and each share is computed as $K_i = P^{y_i}$ in a suitable finite group. Each y_i value is distributed to the i^{th} one of N servers. To authenticate, the client chooses a random secret x, and with each server, sends P^x, retrieves $m_i = (P^x)^{y_i}$, and computes $K_i = m_i^{1/x}$. The client reconstructs K, performs a validation test on K, and uses K to decrypt a private digital signature key U. When the validation test succeeds, the client signs a message with U that contains P^x and optionally other values sent by the client based on incorrect passwords mistakenly entered by the same user in attempting to authenticate. Each server verifies the signed message to authenticate the user, and to forgive the user for some reasonable number of mistakes. With knowledge of valid messages, mistakes and all, the server fine-tunes the accounting of bad access attempts. No single server knows K, P, or any of the K_i shares, and no server receives sufficient information to mount a dictionary attack on K or P. Password security is maintained in a very simple model, requiring no previously secured or server-authenticated channel between the client and any servers. This model further prevents risks inherent in systems where people must authenticate servers, but don't. Data protected by a small password, and no other keys, remains secret even against an enemy that compromises any, but not all, of two or more cooperating authentication servers.